International Journal of Scientific and Innovative Mathematical Research (IJSIMR)
Volume 2, Issue 8, August 2014, PP 705-709
ISSN 2347-307X (Print) & ISSN 2347-3142 (Online)
www.arcjournals.org

Second-Order Rectangular Domain Decomposition for Two-Dimensional Parabolic Problems

Younbae Jun

Department of Applied Mathematics
Kumoh National Institute of Technology
Gumi, Gyeongbuk, Korea
yjun@kumoh.ac.kr

Abstract: *In this paper, an efficient and unconditionally stable rectangular domain decomposition algorithm is proposed. The order of accuracy of the prediction scheme of the new algorithm is second. Numerical experiments support efficiency, accuracy, and unconditional stability of the method.*

Keywords: *Rectangular domain decomposition, parabolic problem, efficiency, unconditional stability.*

1. INTRODUCTION

The quantities such as temperature, velocity, density, pressure, concentration, or electromagnetic field are often governed by a partial differential equation (PDE) of the form

$$F\left(x_1,\cdots,x_n,u,\frac{\partial u}{\partial x_1},\cdots,\frac{\partial u}{\partial x_n},\frac{\partial^2 u}{\partial x_1\partial x_1},\cdots,\frac{\partial^2 u}{\partial x_1\partial x_n},\cdots\right)=0 \tag{1.1}$$

where$x_1,\cdots,x_n$ are independent and u is dependent. For examples, heat equation, wave equation, spherical waves, Laplace equation, Euler-Tricomi equation, advection equation, Ginzburg-Landau equation, Dym equation, vibrating string, and vibrating membrane are explained by PDEs. Many researches have been done to solve the equation (1.1) such as generalized finite difference method [1], implicit collocation method [2], meshless method [3], ADI compact scheme [4], pseudo-spectral method [5], combined spectral method [6], and ADI finite volume method [7].

Domain decomposition (DD) technique is often used for solving the PDE (1.1), too. DD method is very efficient, especially when a parallel computer is used. The basic idea of DD method is that the original spatial domain is decomposed into subdomains and the PDE in each subdomain is solved in parallel manner. Many DD methods were recently proposed, for examples, second-order implicit prediction method [8], explicit/implicit Galerkin method [9], stabilized explicit Lagrange multiplier method [10], and alternating explicit-implicit method [11]. In the case of rectangular domain decomposition, it has been shown that the rectangular modified implicit prediction method [12] is a very efficient unconditionally stable method. However, the prediction scheme of the algorithm is not accurate. In this research, we provide a new rectangular domain decomposition algorithm that improves the accuracy of the prediction scheme of second order.

In this paper, we restrict ourselves to the following two-dimensional parabolic partial differential equation of the form

$$u_t=u_{xx}+u_{yy}+\alpha(x,y)u_x+\beta(x,y)u_y+\gamma(x,y)u+f(x,y,t) \tag{1.2}$$

defined in the unit square $\Omega=[0,1]\times[0,1]$ and $0\le t\le T$, with the initial and Dirichlet boundary conditions

$$u(x,y,0)=u_0(x,y)\text{in }\Omega \tag{1.3}$$

$$u(x,y,t)=u_d(x,y,t)\text{ on the boundary }\partial\Omega, 0\le t\le T \tag{1.4}$$

using a rectangular non-overlapping spatial domain decomposition in Fig. 1.

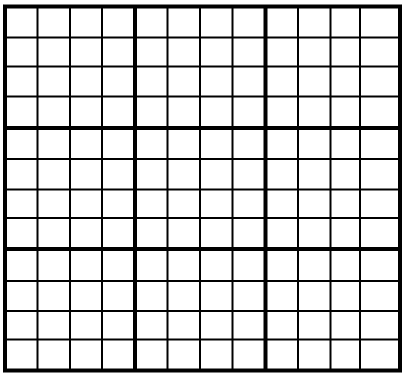

Fig1. *Rectangular decomposition*

2. RECTANGULAR ALGORITHM

In this section, we describe a new rectangular algorithm to solve the initial-boundary value problem $(1.2)-(1.4)$. Finite difference scheme is used to discretize the PDE and the domain of the problem. The positive integers $L, M,$ and N are chosen so that $\Delta x=\frac{1}{L}, \Delta y=\frac{1}{M}$, and $\Delta t=\frac{T}{N}$. Let $x_i=i\Delta x, y_j=j\Delta y,$ and $t_n=n\Delta t$, where $i=0,\cdots,L, j=0,\cdots,M,$ and $n=0,\cdots,N$. Let u_{ij}^n be the exact value $u(x_i,y_j,t_n)$ and w_{ij}^n be the approximated value of u_{ij}^n at the grid point (x_i,y_j,t_n).Then we define the finite difference operators for the time level $t=t_n$ at the point (x_i,y_j) as the followings:

$$w_t^n=\frac{w_{ij}^n-w_{ij}^{n-1}}{\Delta t}, w_{xx}^n=\frac{w_{i+1,j}^n-2w_{ij}^n+w_{i-1,j}^n}{(\Delta x)^2}, w_{yy}^n=\frac{w_{i,j+1}^n-2w_{ij}^n+w_{i,j-1}^n}{(\Delta y)^2},$$

$$w_x^n=\frac{w_{i+1,j}^n-w_{i-1,j}^n}{2\Delta x}, w_y^n=\frac{w_{i,j+1}^n-w_{i,j-1}^n}{2\Delta y}.$$

Let us remark on some methods for solving the initial-boundary value problem $(1.2)-(1.4)$.

Remark 2.1The backward time and centered space (BTCS) method [13] which can be written as the following is a well-known unconditionally stable method:

$$w_t^n=w_{xx}^n+w_{yy}^n+\alpha_{ij}w_x^n+\beta_{ij}w_y^n+\gamma_{ij}w_{ij}^n+f_{ij}^n. \tag{2.1}$$

Note that the BTCS scheme is not a domain decomposition method, but it is used as a benchmark to compare the performances of two different methods.

Remark 2.2 The modified implicit prediction (MIP) method [14] which can be written as the following is an unconditionally stable domain decomposition method:

$$w_t^n=\bar{w}_{xx}^n+w_{yy}^n+\alpha_{ij}\bar{w}_x^n+\beta_{ij}w_y^n+\gamma_{ij}w_{ij}^n+f_{ij}^n,$$

where $\bar{w}_{xx}^n=\frac{2[au_{Lj}^n-(a+b)w_{ij}^n+bu_{oj}^n]}{ab(a+b)H^2}$, $\bar{w}_x^n=\frac{u_{Lj}^n-u_{oj}^n}{(a+b)H}$, $aH=x_i$, and $bH=x_L-x_i$.

In this method, $H(=1/P)$ is the distance of adjacent interface lines if P is the number of decomposed subdomains. The MIP method consists of two components: the prediction step at the interface points and the interior solving step at the interior points. In general, when we solve the parabolic problem $(1.2)-(1.4)$, the BTCS scheme generates a five-diagonal linear system at each time level, but the prediction procedure at the interface line of the MIP method generates only tri-diagonal linear system because of using boundary information at the two-ended grid points. Once the interface values are estimated, the divided smaller problems on each subdomain within adjacent interface lines are then solved by the BTCS scheme independently. Using this domain decomposition technique, the MIP method could be very efficient method to solve the parabolic problem.

Remark 2.3 The rectangular modified implicit prediction (Rectangular MIP) method [8] is an unconditionally stable rectangular domain decomposition method.

In the rectangular MIP algorithm, we have to estimate the values not only at the vertical interface lines but also at the horizontal ones. Thus, the rectangular MIP method consists of three components: the vertical prediction step at the vertical interface points, the horizontal prediction step at the horizontal interface points, and the interior solving step at the interior points.The additional horizontal estimation is the one as follows:

$$w_t^n = w_{xx}^n + \overline{w}_{yy}^n + \alpha_{ij} w_y^n + \beta_{ij} \overline{w}_y^n + \gamma_{ij} w_{ij}^n + f_{ij}^n,$$

where $\overline{w}_{yy}^n = \frac{2[cu_{iM}^n - (c+d)w_{ij}^n + du_{i0}^n]}{cd(c+d)H^2}$, $\overline{w}_y^n = \frac{u_{iM}^n - u_{i0}^n}{(c+d)H}$, $cH = y_j$, and $dH = y_M - y_j$.Even though the rectangular MIP method is very efficient, the estimations of the vertical and horizontal interface values were quite rough. It was desired to improve the accuracy of those estimations in the algorithm. In this paper, we provide a new rectangular algorithm of second order accuracy at the estimations of the vertical and horizontal interface values, which is in cooperation with the second-order implicit prediction domain decomposition (SIPDD) method [8]. Thus, this new rectangular algorithm is referred to as the Rectangular SIPDD method.

We note that the excellent performance of the SIPDD method is reported in [8], but the SIPDD method uses the vertical decomposition only. In this paper, we focus on the performance of rectangular decomposition to see if there is any improvement on the accuracy of the solutions of the parabolic problem $(1.2) - (1.4)$.

Theorem 2.4 The Rectangular SIPDD method is an unconditionally stable rectangular domain decomposition method in which the accuracy at the interface estimations is second.

Proof. Let $\widehat{w}_{xx}^n, \widehat{w}_{yy}^n, \widehat{w}_x$ and $\widehat{w}_y$ be the finite difference operators defined by $\widehat{w}_{xx}^n = \frac{u_{i+LH,j}^n - 2w_{ij}^n + u_{i-LH,j}^n}{H^2}$, $\widehat{w}_{yy}^n = \frac{u_{i,j+MH}^n - 2w_{ij}^n + u_{i,j-MH}^n}{H^2}$, $\widehat{w}_x^n = \frac{u_{i+LH,j}^n - u_{i-LH,j}^n}{2H}$, and $\widehat{w}_y^n = \frac{u_{i,j+MH}^n - u_{i,j-MH}^n}{2H}$, respectively. Then it is easy to see that $\widehat{w}_{xx}^n = u_{xx} + O(H^2)$, $\widehat{w}_{yy}^n = u_{yy} + O(H^2)$, $\widehat{w}_x^n = u_x + O(H^2)$, and $\widehat{w}_y^n = u_y + O(H^2)$. Similar to the BTCS scheme, define the vertical interface prediction operator and the horizontal one by $w_t^n = \widehat{w}_{xx}^n + w_{yy}^n + \alpha_{ij}\widehat{w}_x^n + \beta_{ij} w_y^n + \gamma_{ij} w_{ij}^n + f_{ij}^n$ and $w_t^n = w_{xx}^n + \widehat{w}_{yy}^n + \alpha_{ij} w_x^n + \beta_{ij}\widehat{w}_y^n + \gamma_{ij} w_{ij}^n + f_{ij}^n$, respectively.Once interface points are estimated, the values at the interior points are solved by the classical BTCS scheme. Then the overall rectangular method whose three components are unconditionally stable BTCS methods is unconditionally stable.□

We summarize the rectangular SIPDD algorithm in the following:

<*Rectangular SIPDD algorithm*>

Step1: Predict interface values at the vertical line $x = x_i$ using

$$w_t^n = \widehat{w}_{xx}^n + w_{yy}^n + \alpha_{ij}\widehat{w}_x^n + \beta_{ij} w_y^n + \gamma_{ij} w_{ij}^n + f_{ij}^n,$$

where$\widehat{w}_{xx}^n = \frac{u_{i+LH,j}^n - 2w_{ij}^n + u_{i-LH,j}^n}{H^2}$ and $\widehat{w}_x^n = \frac{u_{i+LH,j}^n - u_{i-LH,j}^n}{2H}$

Step2: Predict interface values at the horizontal line $y = y_j$using

$$w_t^n = w_{xx}^n + \widehat{w}_{yy}^n + \alpha_{ij} w_x^n + \beta_{ij}\widehat{w}_y^n + \gamma_{ij} w_{ij}^n + f_{ij}^n,$$

where$\widehat{w}_{yy}^n = \frac{u_{i,j+MH}^n - 2w_{ij}^n + u_{i,j-MH}^n}{H^2}$ and $\widehat{w}_y^n = \frac{u_{i,j+MH}^n - u_{i,j-MH}^n}{2H}$

Step3: Solve interior linear system using the BTCS scheme (2.1)

$$w_t^n = w_{xx}^n + w_{yy}^n + \alpha_{ij} w_x^n + \beta_{ij} w_y^n + \gamma_{ij} w_{ij}^n + f_{ij}^n$$

Step4: Repeat Step1 through Step3 until the last time level

3. NUMERICAL RESULTS

In this section, we provide numerical results on the performances of the several methods that mentioned earlier in this paper, which are the BTCS scheme, the Rectangular MIP method, and the Rectangular SIPDD method at the various $\lambda = \frac{\Delta t}{(\Delta x)^2} + \frac{\Delta t}{(\Delta y)^2}$. For the rectangular decomposition, we divide the spatial domain into P vertical subdomains and P horizontal subdomains. In this case, the last two methods are referred to as Rectangular MIP($P \times P$) and Rectangular SIPDD($P \times P$), respectively. All of the numerical experiments are carried out on a computer with Intel(R) Core(TM) i7-3770 CPU at 3.40GHz with 4.00GB RAM. In this paper, we use the following two model problems:

(1) Model Problem 1 (MP1):$u_t = u_{xx} + u_{yy} - u + f(x,y,t)$

(2) Model Problem 2 (MP2):$u_t = u_{xx} + u_{yy} + xu_x + yu_y - 3u + f(x,y,t)$

Note that the exact solutions of MP1 and MP2 are $u(x,y,t) = e^{-t}\sin x\cos y$and $u(x,y,t) = e^{x+y-t}$, respectively. All the initial and Dirichlet boundary conditions of the model problems are derived from the exact solutions.

Table I shows the maximum error $||w^N - u^N||_\infty$ at the various$\lambda = \frac{\Delta t}{(\Delta x)^2} + \frac{\Delta t}{(\Delta y)^2}$at $T = 1$. The reason of λ ranging from 512 to 4096 is that most conditionally stable methods will diverse at this high λ. As we can see in Table I, all of the methods: BTCS, Rectangular MIP(4×4) and Rectangular SIPDD(4×4) are unconditionally stable. Note that Rectangular MIP($P \times P$) is not as accurate as BTCS, but the accuracy of Rectangular SIPDD($P \times P$) has been improved. Thus, we can see that the Rectangular SIPDD method is as accurate as the BTCS method.

In Table II, the maximum error and CPU time of Rectangular SIPDD($P \times P$) are presented at the various Pin the case of $\Delta x = \Delta y = \frac{1}{64}$and $\Delta t = \frac{1}{16}$ and $T = 1$.Parallel CPU time (PCPU) is considered as the parameter of efficiency of the method, which divide total CPU time by $P \times P$. It should be pointed out that, when the number of subdomains P increases, the maximum error of SIPDD($P \times P$) remainsalmost the same which is desired, however the parallel CPU time decreases significantly, which is very good.Notice that Rectangular SIPDD(1× 1) is the same as BTCS which is non-domain decomposition method.

Table I. *Maximum error comparison at the various*λ

				MP1			MP2	
$\Delta x(=\Delta y)$	Δt	λ	B	RM	RS	B	RM	RS
1/64	1/2	4096	3.32e-3	6.28e-3	3.38e-3	1.98e-2	2.50e-2	2.03e-2
1/64	1/4	2048	1.47e-3	4.36e-3	1.56e-3	8.76e-3	1.83e-2	9.61e-3
1/64	1/8	1024	6.41e-4	3.42e-3	7.49e-4	3.82e-3	1.52e-2	4.78e-3
1/64	1/16	512	2.55e-4	2.93e-3	3.68e-4	1.53e-3	1.71e-2	2.52e-3

B=BTCS, RM=Rectangular MIP(4×4), RS=Rectangular SIPDD(4×4)

Table II. *Maximum error and CPU time of Rectangular SIPDD at the vario*us*P*

		MP1			MP2	
$P \times P$	1×1	2×2	4×4	1×1	2×2	4×4
Max. Error	2.55e-4	5.20e-4	3.68e-4	1.53e-3	4.29e-3	2.52e-3
TCPU	18.57	8.25	3.71	17.44	8.06	3.63
PCPU	18.57	2.06	0.23	17.44	2.02	0.23

TCPU=Total CPU time, PCPU=Parallel CPU time=TCPU÷ ($P \times P$)

4. Conclusion

In this paper, we presented a new rectangular domain decomposition method for solving two-dimensional parabolic partial differential equations. The rectangular MIP method was not as accurate as the backward time and centered space (BTCS) method. However, the new method that was presented in this paper was as accurate as the BTCS method and the order of accuracy is second. Furthermore, the new method is much faster than the BTCS method.

Acknowledgements

This paper was supported by Research Fund, Kumoh National Institute of Technology.

REFERENCES

[1] J.J. Benito, F. Urena, L. Gravete, Solving parabolic and hyperbolic equations by the generalized finite difference method, J. Comput. Appl. Math. 209 (2007), 208−233.

[2] M. Dehghan, A. Mehebbi, High order implicit collocation method for the solution of two-dimensional linear hyperbolic equation, Numer. Methods Partial Differential Equations 25 (2009), 232−243.

[3] M. Dehghan, A. Shokri, A meshless method for numerical solution of a linear hyperbolic equation with variable coefficients in two space dimensions, Numer. Methods Partial Differential Equations 25 (2009), 494−506.

[4] J. Liu, K. Tang, A new unconditionally stable ADI compact scheme for the two-space-dimensional linear hyperbolic equation, Int. J. Comput. Math. 87 (2010), 2259−2267.

[5] N. Mai-Duy, T. Tran-Cong, An efficient domain-decomposition pseudo-spectral method for solving elliptic differential equations, Commun. Numer. Meth. Engng. 24 (2008), 795−806.

[6] M. Ramezani, M. Dehghan, M. Razzaghi, Combined finite difference and spectral methods for the numerical solution of hyperbolic equation with an integral condition, Numer. Methods Partial Differential Equations 24 (2008), 1−8.

[7] M. Yang, Two-time level ADI finite volume method for a class of second-order hyperbolic problems, Appl. Math. Comput. 215 (2010), 3239−3248.

[8] Y. Jun, T.Z. Mai, Second-order treatment of the interface of domain decomposition method for parabolic problems, Numer. Linear Algebra Appl. 17 (2010), 839−853.

[9] D. Pradhan, N. Nataraj, A. Pani, An explicit/implicit Galerkin domain decomposition procedure for parabolic integro-differential equations, J. Appl. Math. Comput. 28 (2008), 295−311.

[10] Z. Zheng, B. Simeon, L. Petzold, A stabilized explicit Lagrange multiplier based domain decomposition method for parabolic problems, J. Comput. Phys. 227 (2008), 5272−5285.

[11] Y. Zhuang, An alternating explicit-implicit domain decomposition method for the parallel solution of parabolic equations, J. Comput. Appl. Math. 206 (2007), 549−566.

[12] Y. Jun, T.Z. Mai, Rectangular domain decomposition method for parabolic problems, J. Korea Soc. Math. Educ. Ser. B: Pure and Appl. Math. 13 (2006), 281−294.

[13] D.M. Young, Iterative Solution of Large Linear Systems, Dover, New York, 2003.

[14] Y. Jun, T.Z. Mai, ADI method − Domain decomposition, Appl. Numer. Math. 56 (2006), 1092−1107.